# Modular Automata 2 Distributed Petri Nets 4 Synthesis

Philippe Darondeau, Laure Petrucci

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Modular Automata 2 Distributed Petri Nets 4 Synthesis*

Philippe Darondeau , Laure Petrucci

**N°6192**

Mai 2007

Systèmes communicants

Rapport technique

ISSN 0249-0803

# Modular Automata 2 Distributed Petri Nets 4 Synthesis

Philippe Darondeau [*], Laure Petrucci [†]

Systèmes communicants
Projet S4

Rapport technique n˚6192 — Mai 2007 — 18 pages

**Abstract:** Modular automata are symbolic transition systems halfway between families of automata and their synchronized products. They allow for analysis of liveness properties without incurring the state space explosion problem. A modular automaton is composed of modules and a synchronization graph. We consider the problem whether such specifications may be implemented by a distributed Petri net up to language equivalence. The challenge is to avoid computing the state space of the modular automaton to be implemented. We show that this is possible, opening the way to the synthesis of nets from modular specifications.

**Key-words:** modular automaton, synchronized product, distributed Petri net, modular synthesis, regular languages, linear systems

*(Résumé : tsvp)*

[*] Inria, Rennes, <darondeau@irisa.fr>
[†] LIPN, Villetaneuse, <petrucci@lipn.univ-paris13.fr>

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

# De la synthèse des réseaux de Petri distribués à partir des automates modulaires

**Résumé :** Les automates modulaires sont des systèmes de transitions symboliques à mi-chemin entre familles d'automates et produits synchronisés. Les automates modulaires permettent d'analyser les propriétés de vivacité sans encourir le problème de l'explosion combinatoire de l'espace des états. Un automate modulaire se compose de modules et d'un graphe de synchronisation. Nous considérons la question de l'implémentation d'un automate modulaire par un réseau de Petri distribué à équivalence de langages près. Le défi est d'éviter le calcul de l'espace détats de l'automate modulaire. Nous montrons que cela est possible, ouvrant la voie à la synthèse de réseaux à partir d'automates modulaires

**Mots-clé :** automate modulaire, produit synchronisé, réseau de Petri distribué, synthèse modulaire, langages réguliers, systèmes linéaires

# 1 Introduction

The Petri Net Synthesis problem consists in finding whether an automaton or a language may be realized by a Petri net with injectively labelled transitions. The problem was examined first for elementary nets, and it was decided using the *regions* of a graph [10, 11, 4, 8, 5]. The similar problem for P/T-nets was decided in [1] using the extended regions defined in [14] and [3]. The decision procedures have been implemented in the tool Synet [16]. They were adapted to supervisory control and applied to fairly large case studies in [12] and [13]. However, these procedures do not exploit modularity, and they cannot be applied to products of automata without computing beforehand their complete state space, which may be problematic.

Modular automata are concurrent systems halfway between indexed families of automata $(A_i)_{i=1}^n$ and their products $\otimes_i A_i$. It was shown in [6] and [15] that modular automata can be used to analyze liveness properties without incurring the state space explosion problem. We will examine whether this symbolic representation of products can also be used to avoid state space explosion when synthesizing bounded P/T-nets from products of automata up to language equivalence.

Given an alphabet $\Sigma = \cup_{i=1}^n \Sigma_i$ and an indexed family $(A_i)_{i=1}^n$ of automata over the $\Sigma_i$, one can construct *without ever computing any global state space* a modular automaton *MA* whose language is equal to the language of the product automaton $\otimes_i A_i$, i.e. to the mixed product $\otimes_i L(A_i)$. We examine how much space and time can be saved by resorting to this alternative representation of products for deciding whether $\otimes_i L(A_i)$ is the language of a bounded Petri net. We show that space complexity is significantly decreased when resorting to modular automata provided that the following constraint is imposed on P/T-nets: a transition that belongs to a single alphabet $\Sigma_i$ cannot share any input place with a transition in $\Sigma_j$ for $j \neq i$. This definition of distributed P/T-nets is consistent with and extends the definition given in [2]. We show that time complexity decreases drastically when all synchronized transitions in *MA* (or in $\otimes_i A_i$) are reversible.

The remaining sections are organized as follows. Section 2 recalls the region based synthesis of P/T-nets from finite automata. Section 3 introduces modular automata and studies their sets of linear runs. Section 4 presents distributed P/T-nets and distributed regions. Section 5 computes the distributed regions of a modular automaton. Section 6 completes the synthesis procedure. Some proofs not in the text appear in an appendix.

# 2 P/T-net synthesis from finite automata up to language equivalence

The synthesis of P/T-nets from modular automata is grounded on regions of regular languages. We recall the concept of regions of regular languages and their role in the synthesis of general P/T-nets from finite automata.

**Definition 1 (P/T-nets)** *A* P/T-net *is a bi-partite graph $N = (P,T,F)$, where $P$ and $T$ are disjoint sets of vertices, called* places *and* transitions*, respectively, and $F : (P \times T) \cup (T \times P) \to \mathbb{N}$ is a set of directed edges with non-negative integer weights. A* marking *of $N$ is a map $M : P \to \mathbb{N}$. The* state graph *of $N$ is a labelled graph, with markings as vertices, where there is an edge from $M$ to*

*$M'$ with label $t \in T$ (notation: $M[t\rangle M'$) if and only if, for every place $p \in P$, $M(p) \geq F(p,t)$ and $M'(p) = M(p) - F(p,t) + F(t,p)$. The* reachability graph *of an* initialized *P/T-net $\mathcal{N} = (P,T,F,M_0)$ with the* initial marking *$M_0$ is the induced restriction of its state graph on the set of markings that may be reached from $M_0$. The net $\mathcal{N}$ is* finite *if $P$ and $T$ are finite. The net $\mathcal{N}$ is* bounded *if its reachability graph is finite. The* language *$L(\mathcal{N})$ of the net $\mathcal{N}$ is the set of words $t_1 t_2 \ldots t_l \in T^*$ appearing as labels of net firing sequences $M_0[t_1\rangle M_1[t_2\rangle \ldots M_{l-1}[t_l\rangle$.*

The definitions, propositions and algorithms below have been adapted from [1] and [7].

**Definition 2 (regions)** *Let $L \subseteq \Sigma^*$ be a prefix closed language over $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$. A* region *of $L$ is a non-negative integer vector*

$$r = \langle r_{init}, r^\circ\sigma_1, \sigma_1{}^\circ r, \ldots, r^\circ\sigma_k, \sigma_k{}^\circ r\rangle$$

*such that for all $\sigma \in \Sigma$, $w\sigma \in L \Rightarrow r/w \geq r^\circ\sigma$ where $r/w$ is defined inductively on the words of $\Sigma^*$ with $r/\varepsilon = r_{init}$ and $r/w\sigma = r/w - r^\circ\sigma + \sigma^\circ r$. The region $r$ is a* bounded *region of $L$ if there exists $B \in \mathbb{N}$ such that $r/w \leq B$ for all $w \in L$.*

**Proposition 1** *A non-empty prefix closed language $L$ is equal to $L(\mathcal{N})$ for some bounded net $\mathcal{N}$ if and only if, for all $w \in L$ and $\sigma \in \Sigma$, $w\sigma \notin L \Rightarrow r/w < r^\circ\sigma$ for some bounded region $r$ of $L$ (we say that $r$* disables *$\sigma$ after $w$). Let $R$ be any set of bounded regions of $L$ such that, for every minimal word $w\sigma \notin L$, $\sigma$ is disabled after $w$ by some region in $R$. Then $L = L(\mathcal{N})$ where $\mathcal{N} = (P,\Sigma,F,M_0)$ is the bounded P/T-net defined as follows:*
*- $P$ is a set of places $p_r$ in bijective correspondence with the regions $r \in R$,*
*- $M_0(p_r) = r_{init}$ and for any $\sigma \in \Sigma$, $F(p_r,\sigma) = r^\circ\sigma$ and $F(\sigma,p_r) = \sigma^\circ r$.*

When $L = L(A)$ is the language of a finite deterministic automaton $A$, a finite representation of the (infinite) set of bounded regions of $L$ may be computed.

Let $A = (Q,\Sigma,\delta,q_0)$ where $\delta : Q \times \Sigma \rightharpoonup Q$ is a partial map, all states in $Q$ can be reached from $q_0$, and they are all accepting states. Let $\mathcal{U}A$ denote the *finite unfolding* of $A$ constructed as follows. The set of vertices of $\mathcal{U}A$ is the subset of $\Sigma^*$ defined inductively from the empty word $\varepsilon$ (the root vertex) by appending vertex $w\sigma$ as a son of vertex $w$ whenever $\delta(q_0,w\sigma)$ is defined and it differs from $\delta(q_0,v)$ for every prefix $v$ of $w$. The edge from $w$ to $w\sigma$ is labelled with $\sigma$. This yields a finite tree that *spans* $A$. A finite number of *chords* are then added, namely an edge with label $\sigma$ is added from vertex $w$ to vertex $v$ if $\delta(q_0,w\sigma) = \delta(q_0,v)$ and the vertex $v$ is an ancestor of the vertex $w$.

**Proposition 2** *Given $r = \langle r_{init}, r^\circ\sigma_1, \sigma_1{}^\circ r, \ldots, r^\circ\sigma_k, \sigma_k{}^\circ r\rangle$, let $r \times w$ be defined for $w \in \Sigma^*$ inductively with $r \times \varepsilon = 0$ and $r \times w\sigma = r \times w - r^\circ\sigma + \sigma^\circ r$, then $r$ is a bounded region of $L(A)$ if and only if the following conditions hold:*
i) *$r \times u\sigma = 0$ for every chord $vu \xrightarrow{\sigma} v$ in $\mathcal{U}A$,*
ii) *$r/w \geq r^\circ\sigma$ for every edge $w \xrightarrow{\sigma} w'$ in $\mathcal{U}A$.*

**Remark 1** *The above conditions entail that $r \times w = 0$ for every $w$ such that $vw^* \subseteq L(A)$, and $r/w \geq r^\circ\sigma$ for every $w \in L(A)$ such that $w\sigma \in L(A)$.*

Let us now recall two general definitions.

**Definition 3** *Let $G = (V, E, \lambda)$ be a directed graph with sets of vertices and edges $V$ and $E$, respectively, and with an edge labelling map $\lambda : E \rightarrow \Sigma$. A* cycle *in $G$ is an alternated sequence $v_1 e_1 v_2 e_2 \ldots e_p v_1$ where $e_j$ is an edge from $v_j$ to $v_{(j+1) \bmod p}$ and $v_j \neq v_{j'}$ for $j \neq j'$. A* path *in $G$ is an alternated sequence $v_1 e_1 v_2 e_2 \ldots e_p v_{p+1}$ where $e_j$ is an edge from $v_j$ to $v_{j+1}$ and $v_j \neq v_{j'}$ for $j \neq j'$. The* label *of a cycle or path $v_1 e_1 v_2 e_2 \ldots e_p v_{p+1}$ is the word $\lambda(e_1) \ldots \lambda(e_p)$.*

**Definition 4** *The* Parikh image *of a word $w$ of $\Sigma^*$, where $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$, is the map $\psi w : \{1, \ldots, k\} \rightarrow \mathbb{N}$ such that $(\psi w)(j)$ is the number of occurrences of the letter $\sigma_j$ in $w$.*

Thus, each condition of the form $r \times w = 0$ in Prop. 2 is equivalent to a linear homogeneous equation:

$$\Sigma_{h=1}^{k} \vec{X}(h) \times ({\sigma_h}^\circ r - r^\circ \sigma_h) = 0 \tag{1}$$

where $\vec{X}$ is the Parikh image of the label of a cycle in $\mathcal{U}A$. Similarly, each condition of the form $r/w \geq r^\circ\sigma$ in Prop. 2 is equivalent to a linear homogeneous inequality:

$$r_{init} + \Sigma_{h=1}^{k} \vec{Y}(h) \times ({\sigma_h}^\circ r - r^\circ \sigma_h) - (r^\circ \sigma) \geq 0 \tag{2}$$

where $\vec{Y}$ is the Parikh image of the label of a rooted path in $\mathcal{U}A$.

Let $\mathcal{REG}$ denote the resulting system of linear equations (1) and inequalities (2) in the $2k+1$ integer variables $r_{init}$, ${\sigma_h}^\circ r$ and $r^\circ\sigma_h$, augmented with $r_{init} \geq 0$ and with ${\sigma_h}^\circ r \geq 0$ and $r^\circ\sigma_h \geq 0$ for all $h$.

**Proposition 3** *An integer vector $r$ is a bounded region of $L(A)$ if and only if all linear constraints in the finite linear system $\mathcal{REG}$ are satisfied.*

From propositions 1 and 3, $L(A) = L(\mathcal{N})$ for some bounded P/T-net $\mathcal{N}$ if and only if, for each letter $\sigma$ and for each vertex $w$ of $\mathcal{U}A$, either $w \xrightarrow{\sigma} w'$ for some vertex $w'$ or $r/w < r^\circ\sigma$ for some bounded region $r$. Now $r/w < r^\circ\sigma$ is equivalent to the linear homogeneous inequality:

$$r_{init} + \Sigma_{h=1}^{k} \vec{Y}(h) \times ({\sigma_h}^\circ r - r^\circ \sigma_h) - (r^\circ \sigma) < 0 \tag{3}$$

where $\vec{Y}$ is the Parikh image of (the label of the path from the root to) $w$.

Inequality (3) holds for some bounded region $r$ of $L(A)$ if and only if the linear system $\mathcal{REG}$ extended with the inequality:

$$r_{init} + \Sigma_{h=1}^{k} \vec{Y}(h) \times ({\sigma_h}^\circ r - r^\circ \sigma_h) - (r^\circ \sigma) \leq -1 \tag{4}$$

is feasible in $\mathbb{Q}^{2k+1}$, which can be decided in polynomial time. Indeed, rational solutions always induce integer solutions. Let $R$ be any set of bounded regions of $L(A)$ large enough for disabling $\sigma$ after $w$ whenever $\sigma \in \Sigma$, $w$ labels a path from the root of $\mathcal{U}A$, and $w\sigma \notin L(A)$. Then $L(A) = L(\mathcal{N})$ where $\mathcal{N}$ is constructed as indicated in proposition 1.

The number of vertices (resp. edges) of $\mathcal{U}A$ is $1 + |\Sigma|^{|Q|-1}$ (resp. $|\Sigma|^{|Q|}$), hence deciding whether $L(A) = L(\mathcal{N})$ for some $\mathcal{N}$ requires solving in the worst case $O(|\Sigma|^{|Q|})$ linear systems with size $O(|\Sigma|^{|Q|})$.

# 3 Modular Automata

Henceforth, $\Sigma = \Sigma_1 \cup \ldots \cup \Sigma_n = \{\sigma_1, \ldots, \sigma_k\}$, and for any $\sigma$ in $\Sigma$, $dom(\sigma) = \{i \,|\, \sigma \in \Sigma_i\}$. Let $\pi_i : \Sigma^* \to \Sigma_i^*$ be the unique monoid morphism such that $\pi_i(\sigma) = \sigma$ for $\sigma \in \Sigma_i$ and $\pi_i(\sigma) = \varepsilon$ (the empty word) otherwise. Finally let $\Sigma_i' = \Sigma_i \setminus (\cup_{j \neq i} \Sigma_j)$ for $i \in \{1, \ldots, n\}$, and $\Sigma' = \cup_i \Sigma_i'$.

In order to motivate the introduction of modular automata, let us recall two definitions.

**Definition 5 (mixed product of languages [9])** *Given $L_i \subseteq \Sigma_i^*$, $i \in \{1, \ldots, n\}$, let $\otimes_i L_i \subseteq \Sigma^*$ be defined as $\{w \,|\, \forall i : \pi_i(w) \in L_i\}$.*

**Definition 6 (mixed product of automata)** *Given $A_i = (Q_i, \Sigma_i, \delta_i, q_{i,0})$, $i \in \{1, \ldots, n\}$, where $q_{i,0} \in Q_i$ is the initial state, all states in $Q_i$ are accepting, and $\delta_i$ maps partially $Q_i \times \Sigma_i$ to $Q_i$, let $\otimes_i A_i = (Q, \Sigma, \delta, q_0)$ where $q_0 = \langle q_{1,0}, \ldots, q_{n,0} \rangle$, $Q = \{q_0\} \cup range(\delta)$, and $\delta$ and $Q$ are defined inductively from $Q = \{q_0\}$ by adding statements $\delta(\langle q_1, \ldots, q_n \rangle, \sigma) = \langle q_1', \ldots, q_n' \rangle$ for all states $\langle q_1, \ldots, q_n \rangle$ in $Q$ such that $q_i' = \delta_i(q_i, \sigma)$ for $\sigma \in \Sigma_i$ and $q_i' = q_i$ otherwise.*

Thus $L(\otimes_i A_i) = \otimes_i L(A_i)$. Modular automata lay in between families of automata $(A_i)_{i=1}^n$ and their mixed products $\otimes_i A_i$.

**Definition 7 (modular automaton [15])** *A* modular automaton *over $\Sigma$ is a $(n+1)$-tuple $MA = ((A_i')_{i=1}^n, \mathcal{S})$ in which the* modules $A_i' = (Q_i', \Sigma_i', \delta_i', q_{i,0})$ *and the* synchronization graph $\mathcal{S} = (S, T, s_0)$ *comply with the following requirements:*

*- $S \subseteq Q_1' \times \ldots \times Q_n'$ and $s_0 = \langle q_{1,0}, \ldots, q_{n,0} \rangle \in S$,*

*- $T$ is a set of labelled transitions $s \xrightarrow{\langle \phi, \sigma \rangle} s'$ where $s, s'$ are* global states *in $S$, $\sigma$ is a* synchronized action *in $\Sigma \setminus \Sigma'$, and $\phi$ maps $dom(\sigma)$ to* local states *such that $\phi(i) \in Q_i'$,*

*- for each $i$, $\delta_i'$ maps partially $Q_i' \times \Sigma_i'$ to $Q_i'$ — we also denote by $\delta_i'$ the inductive extension of $\delta_i'$ to $Q_i' \times (\Sigma_i')^*$*

*- for any* synchronized transition $\langle q_1, \ldots, q_n \rangle \xrightarrow{\langle \phi, \sigma \rangle} \langle q_1', \ldots, q_n' \rangle$ *and for each $i \leq n$, if $i \notin dom(\sigma)$ then $q_i' = q_i$ else $\phi(i) = \delta_i'(q_i, w_i')$ for some $w_i' \in (\Sigma_i')^*$.*

The intuition underlying the definition of modular automata is as follows. Each transition $\langle q_1, \ldots, q_n \rangle \xrightarrow{\langle \phi, \sigma \rangle} \langle q_1', \ldots, q_n' \rangle$ of the synchronization graph represents a bunch of computations, in which each module $A_i'$ with $i \in dom(\sigma)$ executes on its own a sequence of local transitions from $q_i$ to $\phi(i)$ before *jumping* jointly with the other modules from $\phi(i)$ to $q_i'$ under the effect of the synchronized transition $\sigma$.

In the sequel, every modular automaton $MA = ((A_i')_{i=1}^n, \mathcal{S})$ is assumed to be *finite*, meaning that sets $Q_i'$ and $S$ are finite, and *reachable*, meaning that every vertex $s$ of $\mathcal{S}$ can be reached from the root $s_0$ of this graph and every source state $q_i$ of any module $A_i'$ equals $s(i)$ for some vertex $s \in S$, where $s(i) = q_i$ for $s = \langle q_1, \ldots, q_n \rangle$.

The intuitive view of modular automata suggested above leads to define their semantics, i.e. their language, through expansion to ordinary automata.

**Definition 8** *The* expansion *of the modular automaton* $MA = ((A'_i)_{i=1}^n, \mathcal{S})$ *of Def. 7 is the automaton* $Exp(MA) = (Q, \Sigma, \delta, q_0)$ *where* $Q \subseteq (Q'_1 \times \ldots \times Q'_n) \times S$*,* $q_0 = \langle s_0, s_0 \rangle \in Q$*,* $\delta$ *maps partially* $Q \times \Sigma$ *to* $Q$*, and* $\delta$ *and* $Q$ *are defined inductively (from* $Q = \{q_0\}$ *and* $\delta = \emptyset$*) as follows:*
*for any state* $\langle\langle q_1, \ldots, q_n\rangle, s\rangle$ *in* $Q$*, if condition* (i) *or* (ii) *below holds, then let* $\delta(\langle\langle q_1, \ldots, q_n\rangle, s\rangle, \sigma) = \langle\langle q'_1, \ldots, q'_n\rangle, s'\rangle$ *and* $\langle\langle q'_1, \ldots, q'_n\rangle, s'\rangle \in Q$

i) $\sigma \in \Sigma'_i$ *for some* $i$ *(thus unique),* $s = s'$*,* $q'_i = \delta'_i(q_i, \sigma)$ *and* $q'_j = q_j$ *for* $j \neq i$*,*

ii) $\sigma \in \Sigma \setminus \Sigma'$ *and* $s \xrightarrow{\langle\phi,\sigma\rangle} s'$ *in* $\mathcal{S}$ *for some* $\phi$ *such that* $q_i = \phi(i)$ *and* $q'_i = s'(i)$ *for* $i \in dom(\sigma)$ *and* $q'_i = q_i$ *otherwise.*
*The* language *of the modular automaton is* $L(MA) = L(Exp(MA))$*.*

**Lemma 1** $\forall s \in S\ \exists w \in \Sigma^*\ \delta(\langle s_0, s_0\rangle, w) = \langle s, s\rangle$ *in* $Exp(MA)$*.*

A family of automata $(A_i)_{i=1}^n$ may be transformed into a modular automaton $MA = ((A'_i)_{i=1}^n, \mathcal{S})$ such that $L(MA) = L(\otimes_i A_i)$ as follows. For each $i$, let $A'_i$ be the residue of $A_i$ left after removing all transitions with labels in $\Sigma_i \setminus \Sigma'_i$. For each $i$, let $A''_i$ be produced from $A_i$ as follows: keep all transitions with labels in $\Sigma_i \setminus \Sigma'_i$ and their source and target states, remove all other states and transitions, and add compensatory arcs labelled with $\varepsilon$ to represent sequences of removed transitions. Then $\mathcal{S}$ is just another form of $\otimes_i A''_i$.

Languages of modular automata enjoy useful properties of partial commutation, that we examine now. Henceforth, $q \xrightarrow{w} q'$ is used as another notation for $\delta(q, w) = q'$.

**Definition 9** *Let* $\equiv$ *be the congruence on words of* $\Sigma^*$ *generated from the partial commutation relations* $\sigma\sigma' \equiv \sigma'\sigma$ *if* $(\exists i \in \{1, \ldots, n\})\ \sigma' \in \Sigma'_i \wedge \sigma \notin \Sigma_i$*.*

When two actions commute, at least one belongs to a single alphabet $\Sigma_i$ since it belongs to $\Sigma'_i = \Sigma_i \setminus (\cup_{j \neq i} \Sigma_j)$ while the other does not belong to $\Sigma_i$.

**Lemma 2** *If* $q \xrightarrow{w} q' \in Exp(MA)$ *and* $w \equiv w'$*, then* $q \xrightarrow{w'} q' \in Exp(MA)$*.*

In view of Lemma 2, $L(MA)$ may be described by any set of representatives w.r.t. the congruence $\equiv$. Now, we construct a minimal and regular set of such representatives.

**Definition 10** *For any* $s, s' \in S$*, let* $L(s, s')$ *be the set of the words* $w_1 \ldots w_n \sigma$ *such that* $w_i \in (\Sigma'_i)^*$*,* $\sigma \in \Sigma \setminus \Sigma'$*,* $s \xrightarrow{\langle\phi,\sigma\rangle} s'$ *in* $\mathcal{S}$*, and for all* $i$*, if* $i \notin dom(\sigma)$ *then* $w_i = \varepsilon$ *else* $\phi(i) = \delta'_i(s(i), w_i)$*, where* $s(i) = q_i$ *for* $s = \langle q_1, \ldots, q_n\rangle$*.*

**Lemma 3** $\forall s, s' \in S\ \forall w \in L(s, s')\ \delta(\langle s, s\rangle, w) = \langle s', s'\rangle$ *in* $Exp(MA)$*.*

**Definition 11** *For any* $s = \langle q_1, \ldots, q_n\rangle \in S$*, let* $L(s)$ *be the set of the words* $w_1 \ldots w_n$ *such that* $w_i \in (\Sigma'_i)^*$ *and* $\delta'_i(q_i, w_i)$ *is defined for all* $i \leq n$*.*

**Definition 12** *For* $s, s'$ *ranging over* $S$*, let* $L^+(s, s')$ *be the family of the least languages such that* $L(s, s') \subseteq L^+(s, s')$ *and* $L^+(s, s'')\, L^+(s'', s') \subseteq L^+(s, s')$ *for any* $s'' \in S$*.*

**Definition 13** *Let* $R(MA) = \cup_{s \in S}(L^+(s_0,s)\,L(s)) \cup L(s_0)$.

**Proposition 4** *$R(MA)$ is a minimal set of representatives for $L(MA)$ w.r.t. the congruence $\equiv$ and it is a regular language.*

**Lemma 4** *Let $w, w' \in L(MA)$ and $\alpha\beta, \alpha'\beta' \in R(MA)$ such that $w \equiv \alpha\beta$ and $w' \equiv \alpha'\beta'$, where $\alpha \in L^+(s_0,s)$, $\beta \in L(s)$ and $\alpha' \in L^+(s_0,s')$, $\beta' \in L(s')$. If $w$ is a prefix of $w'$, then $\alpha$ is a prefix of some $\gamma \equiv \alpha'$.*

*Proof.* Up to $\equiv$, the word $\alpha$ (resp. $\alpha'$) is what remains from $w$ (resp. $w'$) after all local actions $\varsigma_i \in \Sigma_i'$ which commute with the first synchronized action on their right have been pushed repeatedly to the right, until entering $\beta$ (resp. $\beta'$). Up to $\equiv$, $\alpha$ must therefore be a prefix of $\alpha'$. ∎

# 4 Distributed P/T-nets and distributed regions

We want to decide, for a modular automaton $MA = ((A_i')_{i=1}^n, \mathcal{S})$, whether $L(MA) = L(\mathcal{N})$ for some bounded P/T-net $\mathcal{N}$. We intend to solve this problem without computing the expansion of $MA$ nor the regular representative set $R(MA)$. More precisely, we aim at deciding whether $L(MA) = L(\mathcal{N})$ by solving in the worst case $K_S \times \Pi_i K_i$ linear systems of size $K_S + \sum_i K_i$ where $K_S = |\Sigma| \times |S|\,!$ and $K_i = |\Sigma_i'|^{|Q_i'|}$. In this evaluation, we assume that $\mathcal{S}$ is simple, i.e. $\delta(s,\langle\phi,\sigma\rangle) = \delta(s,\langle\phi',\sigma'\rangle) \Rightarrow \phi = \phi' \wedge \sigma = \sigma'$, although this may not hold in full generality. Note that $m!$ is an upper bound of the number of paths in a simple automaton with $m$ states. By computing $Exp(MA)$ and applying the synthesis procedure from section 2, one would have to solve $|\Sigma|^q$ linear systems of size $|\Sigma|^q$ where $q = |S| \times \Pi_i |Q_i'|$. Hence, there is a significant gain if $|S|$ is small w.r.t. the $|Q_i'|$, i.e. the system is loosely coupled. This program cannot be carried out unless restrictions are imposed on synthesized nets. We shall restrict ourselves to distributed Petri nets as follows.

**Definition 14 (distributed Petri net)** A distributed Petri net *over $\Sigma = \cup_{i=1}^n \Sigma_i$ is a Petri net $\mathcal{N} = (P,T,F,M_0)$ such that $T = \Sigma$ and $\forall p \in P, \forall i, \forall j : F(p,t') \neq 0$ for $t' \in \Sigma_i'$ ($= \Sigma_i \setminus \cup_{j \neq i} \Sigma_j$) and $i \neq j \implies F(p,t) = 0$ for $t \in \Sigma_j$.*

The intuition underlying this definition is that two transitions $t' = \varsigma_i$ and $t = \varsigma_j$ which are *independent* in the sense that $\varsigma_i\varsigma_j \equiv \varsigma_j\varsigma_i$ (see Def. 9) cannot compete on resource tokens. In case $\Sigma_i \cap \Sigma_j = \emptyset$ for $i \neq j$, this definition coincides with the definition of distributed Petri nets given in [2]. The synthesis of distributed P/T-nets is based on regions as follows.

**Definition 15 (distributed region)** *Let $L$ be a prefix closed language over $\Sigma = \cup_{i=1}^n \Sigma_i$. A* distributed region *of $L$ is a region $r$ of $L$ (see Def. 2) such that $\forall i : r^\circ\sigma' \neq 0$ for $\sigma' \in \Sigma_i' \implies r^\circ\sigma = 0$ for all $\sigma \notin \Sigma_i$.*

The following is a straightforward adaptation of Prop. 1.

**Proposition 5** *A non-empty prefix closed language $L$ is equal to $L(\mathcal{N})$ for some distributed and bounded net $\mathcal{N}$ if and only if, for any $w \in L$ and $\sigma \in \Sigma$, $w\sigma \notin L \Rightarrow r/w < r^\circ\sigma$ for some distributed and bounded region $r$ of $L$. Let $R$ be any set of distributed and bounded regions of $L$ large enough to disable after $w$ every minimal word $w\sigma \notin L$. Then $L = L(\mathcal{N})$ where the distributed and bounded P/T-net $\mathcal{N} = (P, \Sigma, F, M_0)$ is defined from $R$ as in Prop. 1.*

# 5 Computing distributed regions

The next stage is to construct, from $MA = ((A'_i)_{i=1}^n, \mathcal{S})$, a linear characterization of the set of distributed and bounded regions of $L(MA)$.
From remark 1, a non-negative vector $r = \langle r_{init}, r^\circ\sigma_1, \sigma_1{}^\circ r, \ldots, r^\circ\sigma_k, \sigma_k{}^\circ r\rangle$ is a region of $L(MA)$ if and only if:
*i)* $r \times w = 0$ for all $w$ in $\Sigma^*$ such that $vw^* \subseteq L(MA)$ for some $v$,
*ii)* $r/w \geq r^\circ\sigma$ for every $w \in L(MA)$ such that $w\sigma \in L(MA)$.
We construct from $MA$ two linear systems $\mathcal{REG}_1$ and $\mathcal{REG}_2$, equivalent to the predicates *(i)* and *(ii)* respectively for distributed regions $r$.

## 5.1 The linear equations in $\mathcal{REG}_1$

**Lemma 5** *Condition* (i) *holds for all $w \in \Sigma'^*$ such that $vw^* \subseteq L(MA)$ for some $v$, if and only if $r \times w = 0$ for every word $w$ labelling a* local cycle *i.e. a cycle in some module $A'_i$.*

*Proof.* Since all states in modular automata are reachable, *(i)* entails that $r \times w = 0$ if $w$ labels a local cycle. Conversely, suppose that $r \times w = 0$ for all labels of local cycles $w$. Let $v \in \Sigma^*$ and $w \in (\Sigma')^*$ ($w$ contains no synchronized action). Assume that $vw^* \subseteq L(MA)$. Let $\delta(\langle s_0, s_0\rangle, v) = \langle\langle q_1, \ldots, q_n\rangle, s\rangle$ in $Exp(MA)$, and for all $i \in \{1, \ldots, n\}$ let $w_i$ be the projection of $w$ on $(\Sigma'_i)^*$. Clearly $vw_i^* \subseteq L(MA)$ and for all $j$, $\delta(\langle s_0, s_0\rangle, v(w_i)^j) = \langle\langle q_1, \ldots, \delta'_i(q_i, (w_i)^j), \ldots, q_n\rangle, s\rangle$. Since module $A'_i$ has a finite number of states, $\delta'_i(q_i, (w_i)^j) = \delta'_i(q_i, (w_i)^h)$ for some $h > j$. Choose $h$ as small as possible, then $(w_i)^{h-j}$ labels a cycle in $A'_i$, hence $r \times (w_i)^{h-j} = r \times w_i = 0$. As $r \times w = r \times w_1 + \ldots + r \times w_n$, we have $r \times w = 0$. ∎

**Lemma 6** *Condition* (i) *holds if and only if $r \times w = 0$ for every word $w$ labelling a local cycle and for every word $w$ in $L^+(s,s)$ for some $s \in S$.*

*Proof.* Since all states in the synchronization graph $\mathcal{S}$ are reachable, *(i)* entails that $r \times w = 0$ if $w \in L^+(s,s)$ for some $s \in S$. Conversely, assume that $r \times w = 0$ for all words $w$ labelling local cycles or belonging to $L^+(s,s)$ for some $s \in S$, and let $vw^* \subseteq L(MA)$ where $w$ contains at least one synchronized action. As $Exp(MA)$ is a finite automaton, $\delta(\langle s_0, s_0\rangle, vw^j) = \delta(\langle s_0, s_0\rangle, vw^h)$ for some $j < h$ (as depicted in figure 1). We will show $r \times w = 0$ or yet equivalently $r \times w^{h-j} = 0$. Up to renaming $vw^j$ into $v'$ and $w^{h-j}$ into $w'$, we can suppose w.l.o.g. that $v$ contains at least one synchronized action and $\delta(\langle s_0, s_0\rangle, v) = \delta(\langle s_0, s_0\rangle, vw)$. Let $\delta(\langle s_0, s_0\rangle, v) = \langle\langle q_1, \ldots, q_n\rangle, s\rangle$, then by Prop. 4 and Lemma 2, $v \equiv \alpha_1\beta_1 \in R(MA)$ with $\alpha_1 \in L^+(s_0, s)$ and $\beta_1 \in L(s)$. Similarly, for all $i \geq 2$, $vw^{i-1} \equiv \alpha_i\beta_i \in R(MA)$ with $\alpha_i \in L^+(s_0, s)$ and $\beta_i \in L(s)$. As $v < vw < vw^2 \ldots$ is a chain in the prefix order, by Lemma 4, one may choose $\alpha'_i \equiv \alpha_i$ such that $\alpha'_1 < \alpha'_2 \ldots$ is a chain, and w.l.o.g. one may assume $\alpha'_i = \alpha_i$. By Lemma 3, $\delta(\langle s_0, s_0\rangle, \alpha_i) = \langle s, s\rangle$ for all $i$, hence there exists a family of words $\gamma_i \in L^+(s,s)$ such that $\alpha_i = \alpha_1\gamma_1 \ldots \gamma_{i-1}$ for all $i \geq 2$ (see Fig. 1). As $\beta_j w^{h-j} \equiv \gamma_j \ldots \gamma_{h-1}\beta_h$ for all $j < h$, these two words have the same Parikh image, i.e. $\psi\beta_j + (h-j) \times \psi w = \psi\gamma_j + \ldots + \psi\gamma_{h-1} + \psi\beta_h$. Since $\gamma_i \in L^+(s,s)$, $r \times \gamma_i = 0$ for all $i$ by assumption, and $(r \times \beta_j) + (h-j)(r \times w) = (r \times \beta_h)$ holds. In order to prove that $r \times w = 0$, it suffices to show that $r \times \beta_j = r \times \beta_h$ for some $j < h$. This must be the case: the assumption that $r \times \gamma = 0$ for every local cycle $\gamma$ entails that the set of possible values of $r \times \beta$ for $\beta \in L(s)$ is finite (for a fixed $r$). $\blacksquare$

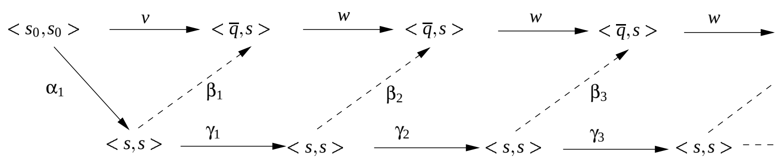


Figure 1:

All labels of local cycles are also labels of cycles in $Exp(MA)$ but seeing that a vertex cannot occur twice in a cycle (Def. 3), some words $w \in L^+(s,s)$ may not correspond to any cycle in $Exp(MA)$. In that case, the constraint $r \times w = 0$ may be discarded because it is redundant with similar constraints for shorter words as we show now.

Let $w = \alpha_1 \ldots \alpha_p \in L^+(s,s)$ where $\alpha_j = w_{j1} \ldots w_{jn}\varsigma_j \in L(s,s)$ for all $j$, with $w_{ji} \in \Sigma_i'^*$ for $i \in dom(\varsigma_j)$ and $w_{ji} = \varepsilon$ otherwise. In view of Def. 8 and Def. 10, there exists in $S$ a sequence of states $s = s_1, s_2, \ldots, s_p, s_{p+1} = s$ such that $\delta(\langle s_j, s_j\rangle, \alpha_j) = \langle s_{j+1}, s_{j+1}\rangle$ for all $j \leq p$. In order that $w$ labels a cycle in $Exp(MA)$, it is necessary that $s_j \neq s_h$ for $j \neq h$ and $\delta'_i(s_j(i), u) \neq \delta'_i(s_j(i), uv)$ for distinct prefixes $u$ and $uv$ of $w_{ji}$ ($j \leq p$ and $i \in dom(\varsigma_j)$). If $s_j = s_h$ for $j < h$ then $r \times w = 0$ comes as a consequence of $r \times w' = 0$ and $r \times w'' = 0$ for shorter words $w' = \alpha_j \ldots \alpha_{h-1}$ and $w'' = \alpha_1 \ldots \alpha_{j-1}\alpha_h \ldots \alpha_p$ in $L^+(s,s)$. If $w_{ji} = uvu'$ with $\delta'_i(s_j(i), u) = \delta'_i(s_j(i), uv)$ then $r \times w$ comes as a consequence of $r \times v = 0$ ($v$ is a concatenation of labels of local cycles) and $r \times w' = 0$ for the shorter word $w' = w[v/\varepsilon]$ in $L^+(s,s)$. Moreover if $\delta'_i(s_j(i), w_{ji}) = \delta'_i(s_j(i), w'_{ji})$ for some $j \leq p$ and $i \in dom(\varsigma_j)$ and we let $w' = w[w_{ji}/w'_{ji}]$, then $w' \in L^+(s,s)$, and $r \times w = 0 \Rightarrow (r \times w' = 0 \Leftrightarrow r \times w_{ji} = r \times w'_{ji})$. This motivates the following definition.

**Definition 16** *For any transition $s \xrightarrow{\langle\phi,\sigma\rangle} s'$ in $\mathcal{S}$ and for any $i \in dom(\sigma)$, if this transition occurs on some cycle in $\mathcal{S}$, or $s(i)$ and $\phi(i)$ occur jointly on some cycle in module $A'_i$, then let $L(s(i), \phi(i))$ be the set of words labelling* paths *from $s(i)$ to $\phi(i)$ in $A'_i$, and let $w(s(i), \phi(i))$ be a distinguished word in $L(s(i), \phi(i))$.*

**Definition 17** *Given a cycle $C = s_1 \xrightarrow{\langle\phi_1,\varsigma_1\rangle} s_2 \ldots \xrightarrow{\langle\phi_p,\varsigma_p\rangle} s_1$ in $\mathcal{S}$, let $\theta(C) = (w_{11} \ldots w_{1n}\varsigma_1) \ldots (w_{p1} \ldots w_{pn}\varsigma_p)$ where for all $i \leq n$ and $j \leq p$, $w_{ji} = \varepsilon$ if $i \notin dom(\varsigma_j)$ and $w_{ji} = w(s_j(i), \phi_j(i))$ otherwise.*

**Definition 18** *Let $\mathcal{REG}_1$ be the linear system in the variables $r^\circ\sigma_h$ and $\sigma_h{}^\circ r$ ($1 \le h \le k$) with one equation:*

$$\sum_{h=1}^{k} (\psi\beta)(h) \times (\sigma_h{}^\circ r - r^\circ\sigma_h) = 0$$

*per $\beta$ equal either to the label of a cycle in some module $A'_i$ or to $\theta(C)$ for some cycle $C$ in $\mathcal{S}$, plus one equation*

$$\sum_{h=1}^{k} (\psi\beta - \psi\beta')(h) \times (\sigma_h{}^\circ r - r^\circ\sigma_h) = 0$$

*for all $\beta = w(s(i), \phi(i))$ and $\beta' \in L(s(i), \phi(i))$ such that $s \xrightarrow{\langle\phi,\sigma\rangle} s'$ occurs in some cycle in $\mathcal{S}$.*

In view of Lemma 6, one can state the following result without further proof.

**Proposition 6** *Let $r$ be a vector in $\mathbb{N}^{2k+1}$ then $r \times w = 0$ for all $w \in \Sigma^*$ such that $vw^* \subseteq L(MA)$ if and only if all constraints in $\mathcal{REG}_1$ are satisfied.*

If the synchronization graph is simple, the number of cycles in $\mathcal{S}$, resp. in $A'_i$, is bounded by the factorial of $|S|$, resp. by $K_i = |\Sigma'_i|^{|Q'_i|}$. The sum of the sizes of all sets $L(s(i), \phi(i))$ for a fixed $i$ is also bounded by $K_i$. Therefore, the number of equations in $\mathcal{REG}_1$ is $O(\,|S|\,! + \sum_i K_i)$.

## 5.2 The linear inequalities in $\mathcal{REG}_2$

We should construct a linear system $\mathcal{REG}_2$ such that, for any integer vector $r$ satisfying $\mathcal{REG}_1$ and $\mathcal{REG}_2$, $r/w \ge r^\circ\sigma$ for all $w$ such that $w\sigma \in L(MA)$. In view of Lemma 2 and Prop. 4, it suffices to check this condition for the words $w \in R(MA)$. Assuming that $r$ satisfies all equations in $\mathcal{REG}_1$, one may further discard all words $w \in R(MA)$ that contain some factor either labelling a cycle in some module $A'_i$ or within $L^+(s,s)$ for some state $s \in S$. Let $IR(MA) \subseteq R(MA)$ be the subset of *irreducible* words obtained in this manner. Clearly, $IR(MA)$ is a *finite* and prefix-closed set of words.

Let $w = (w_{11} \dots w_{1n}\varsigma_1) \dots (w_{p1} \dots w_{pn}\varsigma_p)\, w'_1 \dots w'_n$ be any word in $IR(MA)$, with $w'_i \in \Sigma'^*_i$ for all $i \le n$, and let $\sigma \in \Sigma$ such that $w\sigma \in L(MA)$. Assume that $r^\circ\varsigma \ne 0$ for $\varsigma \in \Sigma'_i$ entails $r^\circ\varsigma' = 0$ for $\varsigma' \notin \Sigma_i$. If $r^\circ\sigma = 0$, then $r/w \ge r^\circ\sigma$ comes as a consequence of $r/v \ge r^\circ\varsigma$ for $w = v\varsigma$. If $r^\circ\sigma \ne 0$, then by the above assumption, $r^\circ\varsigma = 0$ for all actions $\varsigma \in \Sigma'_i$ with $i \notin dom(\sigma)$. Therefore, $r/w \ge r/v$ where $v$ is defined from $w$ by replacing $w'_i$ with $\varepsilon$ for all $i \notin dom(\sigma)$. Let $w'$ be a concatenation of the removed $w'_i$, then $w\sigma \equiv v\sigma w' \in L(MA)$ (Lemma 2), hence $v\sigma \in L(MA)$. Moreover, $v \in IR(MA)$ and $r/v \ge r^\circ\sigma \Rightarrow r/w \ge r^\circ\sigma$. Therefore, a non-negative integer vector $r$ satisfying $\mathcal{REG}_1$ is a distributed region of $L(MA)$ iff $w\sigma \in L(MA) \Rightarrow r/w \ge r^\circ\sigma$ for all irreducible representatives $w \in IR(MA)$.

Words $w$ in $IR(MA)$ are produced from paths $s_0 \xrightarrow{\langle\phi_1,\varsigma_1\rangle} s_1 \;\; \dots \;\; \xrightarrow{\langle\phi_p,\varsigma_p\rangle} s_p$ in $\mathcal{S}$ by choosing for each $j \le p$ and $i \in dom(\varsigma_p)$ the label $w_{ji}$ of some path from $s_{j-1}(i)$ to $\phi_j(i)$ in $A'_i$ and for each $i \le n$ the label $w'_i$ of some path rooted at $s_p(i)$ in $A'_i$, thus yielding $w = (w_{11} \dots w_{1n}\varsigma_1) \dots (w_{p1} \dots w_{pn}\varsigma_p)\, w'_1 \dots w'_n$. If $w\sigma \in L(MA)$ and $w'_i = \varepsilon$ for all $i \notin dom(\sigma)$, then $r/w \geq r^\circ\sigma$ may be equivalently formulated as:

$$r_{init} + \sum_{j=1}^{p} \left( (\varsigma_j \times r) + \sum_{i \in dom(\varsigma_j)} (w_{ji} \times r) \right) + \sum_{i \in dom(\sigma)} (w'_i \times r) \geq r^\circ\sigma \tag{5}$$

In order for $r$ to be a distributed region of $L(MA)$, inequality 5 should hold for arbitrary words $w_{ji}$ labelling paths from $s_{j-1}(i)$ to $\phi_j(i)$ in $A'_i$. If proceeding naively, the number of occurrences of inequality 5 to consider for a single path in $\mathcal{S}$ is the product of all numbers of paths from $s_{j-1}(i)$ to $\phi_j(i)$ for $j \leq p$. This can be avoided by writing for each path in $\mathcal{S}$ a single inequality where $(w_{ji} \times r)$ is replaced with a new variable $z(s_{j-1}(i), \phi_j(i))$ and adding constraints $z(s_{j-1}(i), \phi_j(i)) \leq (w_{ji} \times r)$ independently for all $i$, $j$ and path labels $w_{ji}$. If $w(s_{j-1}(i), \phi_j(i))$ is defined (see Def. 16), these inequalities are replaced with equations $z(s_{j-1}(i), \phi_j(i)) = w(s_{j-1}(i), \phi_j(i)) \times r$.

In order that $r$ be a distributed region of $L(MA)$, inequality 5 should hold either, for $\sigma \in \Sigma'_i$, for all $w'_i \in \Sigma'^*_i$ such that $\delta'_i(s_p(i), w'_i\sigma)$ is defined or, for $\sigma \in \Sigma \setminus \Sigma'$, for all $w'_i \in \Sigma'^*_i$, $i \in dom(\sigma)$, such that $\phi(i) = \delta'_i(s_p(i), w'_i)$ is defined and $\delta(s_p, \langle\phi, \sigma\rangle)$ is defined. In the latter case, inequality 5 should hold for all words $w'_i$ labelling paths from $s_p(i)$ to $\phi(i)$ in $A'_i$. One can avoid writing as many inequalities 5 as choices for the $w'_i$ ($i \in dom(\sigma)$) by replacing $(w'_i \times r)$ with a variable $z(s_p(i), \phi(i))$ and adding constraints $z(s_p(i), \phi(i)) \leq (w'_i \times r)$ as above. In the former case ($\sigma \in \Sigma'_i$), one can similarly avoid writing as many inequalities 5 as choices for $w'_i$ by replacing $(w'_i \times r)$ with a new variable $z(s_p(i), \sigma)$ and adding independent constraints $z(s_p(i), \sigma) \leq (w'_i \times r)$ for all path labels $w'_i$ such that $\delta'_i(s_p(i), w'_i\sigma)$ is defined.

**Definition 19** *Let $V$ be a set of variables comprising $r_{init}$, all $r^\circ\sigma_h$ and $\sigma_h{}^\circ r$ for $1 \leq h \leq k$, all $z(s(i), \phi(i))$ such that $\delta(s, \langle\phi, \sigma\rangle)$ is defined for some $\sigma \in \Sigma \setminus \Sigma'$ and $i \in dom(\sigma)$, and all $z(s(i), \sigma)$ such that $s \in S$, $\sigma \in \Sigma'_i$, and $\delta'_i(s(i), w'_i\sigma)$ is defined for some $w'_i$.*

**Definition 20** *Let $\mathcal{R}EG2$ be the linear system with variables in $V$ constructed as follows:*

- *for each variable $z(s(i), \phi(i))$,*
  - *if $w(s(i), \phi(i))$ is defined (see Def. 16), set*

$$z(s(i), \phi(i)) = \sum_{h=1}^{k} (\psi w(s(i), \phi(i)))(h) \times (\sigma_h{}^\circ r - r^\circ\sigma_h)$$

  - *otherwise, for each word $w$ labelling a path from $s(i)$ to $\phi(i)$ in $A'_i$, set*

$$z(s(i), \phi(i)) \leq \sum_{h=1}^{k} (\psi w)(h) \times (\sigma_h{}^\circ r - r^\circ\sigma_h)$$

- *for each variable $z(s(i), \sigma)$*

– *for each path label $w'_i$ in $A'_i$ such that $\delta'_i(s(i), w'_i\sigma)$ is defined, set*

$$z(s(i),\sigma) \leq \sum_{h=1}^{k} (\psi w'_i)(h) \times (\sigma_h{}^\circ r - r^\circ \sigma_h)$$

- *for each path $s_0 \xrightarrow{\langle \phi_1, \varsigma_1 \rangle} s_1 \ldots \xrightarrow{\langle \phi_p, \varsigma_p \rangle} s_p$*
  – *for all $i \in \{1, \ldots, n\}$ and $\sigma \in \Sigma'_i$ such that $\delta'_i(s_p(i), w'_i\sigma)$ is defined for some $w'_i \in \Sigma'^*_i$ set the inequality*

$$r_{init} + \sum_{j=1}^{p} \left( \varsigma_j{}^\circ r - r^\circ \varsigma_j + \sum_{i \in dom(\varsigma_j)} z(s_{j-1}(i), \phi_j(i)) \right) + z(s_p(i), \sigma) \ \geq \ r^\circ \sigma$$

  – *for any $\varsigma_{p+1} \in \Sigma \setminus \Sigma'$ and any map $\phi_{p+1}$ such that $\delta(s_p, \langle \phi_{p+1}, \varsigma_{p+1} \rangle)$ is defined, set the inequality*

$$r_{init} + \sum_{j=1}^{p+1} \left( \varsigma_j{}^\circ r - r^\circ \varsigma_j + \sum_{i \in dom(\varsigma_j)} z(s_{j-1}(i), \phi_j(i)) \right) \geq \varsigma_{p+1}{}^\circ r$$

In view of the analysis carried out in the beginning of the section, the following result can be stated without further proof.

**Proposition 7** *Let $r$ be a vector in $\mathbb{N}^{2k+1}$, such that $r^\circ \varsigma \neq 0$ for $\varsigma \in \Sigma'_i$ entails $r^\circ \varsigma' = 0$ for $\varsigma' \notin \Sigma_i$. Then $r$ is a bounded and distributed region of $L(MA)$ if and only if it is the projection of a non-negative integer vector satisfying all constraints in $\mathcal{REG}_1 \cup \mathcal{REG}_2$.*

Altogether, we have obtained a linear characterization of the set of bounded and distributed regions of $L(MA)$ with a number of equations or inequalities varying as $O(K_S + \sum_i K_i)$ where $K_S = |\Sigma| \times |S|\,!$ and $K_i = |\Sigma'_i|^{\,|Q'_i|}$.

# 6 The distributed P/T-net synthesis procedure

In order to decide whether $L(MA) = L(\mathcal{N})$ for some bounded and distributed P/T-net $\mathcal{N}$, one should determine whether, for all words $w \in L(MA)$ and actions $\sigma \in \Sigma$ such that $w\sigma \notin L(MA)$, $r/w < r^\circ \sigma$ for some bounded and distributed region $r$ of $L(MA)$. In view of Lemma 2 and Prop. 4, it suffices to consider representative words $w \in R(MA)$. Let $w \in R(MA)$ and suppose $w = \alpha\beta\gamma$ such that, in $Exp(MA)$:
- $\delta(\langle s_0, s_0 \rangle, \alpha) = \langle s, s \rangle$ and $\beta \in L^+(s,s)$, or
- $\delta(\langle s_0, s_0 \rangle, \alpha) = \langle \langle q_1, \ldots, q_n \rangle, s \rangle$, $\beta \in \Sigma'^*_i$ and $\delta'_i(q_i, \beta) = q_i$.
Let $w' = \alpha\gamma$ then $w\sigma \in L(MA) \Leftrightarrow w'\sigma \in L(MA)$ and in both cases $r/w = r/w'$ for any bounded and distributed region $r$ of $L(MA)$ (by Lemma 6 and Prop. 7). Therefore, it suffices to test whether $\sigma$ is disabled after $w$ by some region of $L(MA)$ for irreducible words $w \in IR(MA)$ (such that $w\sigma \notin L(MA)$). A naive synthesis procedure may be defined as follows.

**Algorithm 1**

- *For any path* $s_0 \xrightarrow{\langle\phi_1,\varsigma_1\rangle} s_1 \ldots \xrightarrow{\langle\phi_p,\varsigma_p\rangle} s_p$ *with target* $s_p = \langle q_1, \ldots, q_n\rangle$
- *for any words* $w_{ji}$ *(*$j \le p$*,* $i \in dom(\varsigma_j)$*) labelling paths from* $s_{j-1}(i)$ *to* $\phi_j(i)$
- *for any words* $w'_i$ *(*$i \le n$*) labelling paths from* $q_i$ *to* $\delta'_i(q_i, w'_i)$
- *for any* $\sigma \in \Sigma$
  - *if* $\sigma \in \Sigma'_h$ *and* $\delta'_h(q_h, w'_h\sigma)$ *is undefined,*
  - *or* $\sigma \in \Sigma \setminus \Sigma'$ *and* $\delta(s_p, \langle\phi,\sigma\rangle)$ *is undefined for the map* $\phi$ *defined with* $\phi(i) = \delta'_i(s_p(i), w'_i)$ *for* $i \in dom(\sigma)$
- *solve the linear system formed of* $\mathcal{R}EG1$*,* $\mathcal{R}EG2$*,* $r^\circ\varsigma = 0$ *for all* $\varsigma \notin dom(\sigma)$*, and the inequality:*

$$r_{init} + \sum_{j=1}^{p}\left((\varsigma_j \times r) + \sum_{i\in dom(\varsigma_j)}(w_{ji}\times r)\right) + \sum_{i=1}^{n}(w'_i \times r) < r^\circ\sigma \qquad (6)$$

**Proposition 8** *$L(MA) = L(\mathcal{N})$ for some bounded and distributed Petri net $\mathcal{N}$ if and only if all linear systems in algorithm 1 are feasible. Let R be any set of regions of $L(MA)$ containing at least one solution of each system, then $L(MA) = L(\mathcal{N})$ where $\mathcal{N}$ is the P/T-net constructed from R as indicated in proposition 1.*

*Proof.* Straightforward from Prop. 5 in view of the analysis conducted in the beginning of this section. ∎

The number of linear systems to be solved is a problem, hence one can try saving computational effort by discarding redundant instances of constraint 6. A first cause of redundancy is when $w'_i$ does not label a maximal path in $A'_i$ for some $i \notin dom(\sigma)$. To see this, let $w\sigma \notin L(MA)$ for some word $w = \alpha w'_1 \ldots w'_n$ where $\alpha \in L^+(s_0, s_p)$ and for all $i$, $w'_i$ labels a path from $s_p(i)$ to $\delta'_i(s_p(i), w'_i)$. Consider another word $v = \alpha w''_1 \ldots w''_n$ where for all $i$, $w''_i = w'_i\beta_i$ labels a path from $s_p(i)$ to $\delta'_i(s_p(i), w''_i)$, and $\beta_i = \varepsilon$ for $i \in dom(\sigma)$. In view of Lemma 2, since $v\sigma \equiv w\sigma\beta_1 \ldots \beta_n$, $L(MA)$ is closed under prefix, and $w\sigma \notin L(MA)$, $v\sigma \notin L(MA)$. We claim that any distributed region $r$ which disables $\sigma$ after $v$ disables also $\sigma$ after $w$. As $v \in L(MA)$, $r/v \ge 0$ and $r/v < r^\circ\sigma$ entails $r^\circ\sigma \ne 0$. As $r$ is a distributed region and $r^\circ\sigma \ne 0$, $r^\circ\sigma' = 0$ for any $\sigma' \in \Sigma'_i$ with $i \notin dom(\sigma)$. Therefore, $r \times \beta_i \ge 0$ for all $i$. As $r/v = r/w + \sum_i r \times \beta_i$, the relation $r/v < r^\circ\sigma$ entails $r/w < r^\circ\sigma$. Therefore, it suffices to consider the instances of constraint 6 where, for all $i \notin dom(\sigma)$, the $w'_i$ label maximal paths from $s_p(i)$ to $\delta'_i(s_p(i), w'_i)$.

A second cause of redundancy is when, for some $i \in dom(\sigma)$, $s_p(i)$ and $\delta'_i(s_p(i), w'_i)$ occur jointly on some cycle in $A'_i$, and constraint 6 has already been solved with $w'_i$ changed into $w''_i$ such that $\delta'_i(s_p(i), w'_i) = \delta'_i(s_p(i), w''_i)$. It follows from the equations in $\mathcal{R}EG1$ that $r \times w'_i$ and $r \times w''_i$ are equal in this case. A third cause of redundancy is when, for some $i$ and $j$, $w(s_{j-1}(i), \phi_j(i))$ is defined (see Def. 16) and $w_{ji} \ne w(s_{j-1}(i), \phi_j(i))$. It follows from the equations in $\mathcal{R}EG1$ that $r \times w_{ji} =$

$r \times w(s_{j-1}(i), \phi_j(i))$ in this case. Last but not least, it should be observed that relation 6 depends only on the Parikh images of the factors $w_{ji}$ and $w'_i$ (because $r \times w = \sum_{h=1}^{k} (\psi w)(h) \times (\sigma_h{}^\circ r - r^\circ \sigma_h)$), and this is an important source of redundancy. Taking all possible optimizations into account, one obtains the algorithm described below.

**Definition 21** *For any state $s \in S$ and for any $i \leq n$, let $PIMP(s(i))$ be the set of Parikh images of (words labelling) maximal paths from $s(i)$ in $A'_i$. For any transition $\delta(s, \langle \phi, \sigma \rangle) = s'$ in $\mathcal{S}$ and $i \in dom(\sigma)$, let $PIP(s(i), \phi(i)) = \{\psi w(s(i), \phi(i))\}$ if $w(s(i), \phi(i))$ is defined (see Def. 16), else let $PIP(s(i), \phi(i))$ be the set of Parikh images of all words labelling paths from $s(i)$ to $\phi(i)$ in $A'_i$.*

**Algorithm 2**

- *For any path $s_0 \xrightarrow{\langle \phi_1, \varsigma_1 \rangle} s_1 \ldots \xrightarrow{\langle \phi_p, \varsigma_p \rangle} s_p$ with target $s_p = \langle q_1, \ldots, q_n \rangle$*
- *for any vectors $\vec{x}_{ji} \in PIP(s_{j-1}(i), \phi_j(i))$, $j \leq p$ and $i \in dom(\varsigma_j)$,*
- *for any $\sigma \in \Sigma$*
- *for any vectors $\vec{x'}_i \in PIMP(q_i)$ for $i \notin dom(\sigma)$*
- *for any words $w'_i$ labelling paths from $q_i$ to $\delta'_i(q_i, w'_i)$ for $i \in dom(\sigma)$*
  - *if $\sigma \in \Sigma'_h$ and $\delta'_h(q_h, w'_h \sigma)$ is undefined,*
  - *or $\sigma \in \Sigma \setminus \Sigma'$ and $\delta(s_p, \langle \phi, \sigma \rangle)$ is undefined for $\phi(i) = \delta'_i(s_p(i), w'_i)$,*
- *solve the linear system formed of $\mathcal{R}EG1$, $\mathcal{R}EG2$, $r^\circ \varsigma = 0$ for all $\varsigma \notin dom(\sigma)$, and the equations:*

$$\vec{x} = \sum_{j=1}^{p} \left( (\psi \varsigma_j) + \sum_{i \in dom(\varsigma_j)} \vec{x}_{ji} \right) + \sum_{i \notin dom(\sigma)} \vec{x'_i} + \sum_{i \in dom(\sigma)} \psi w'_i \tag{7}$$

$$r_{init} + \sum_{h=1}^{k} \vec{x}(h) \times (r^\circ \varsigma_h - \varsigma_h r^\circ) < r^\circ \sigma \tag{8}$$

In spite of the optimization, the number of linear systems to be solved is the same in the worst case as with algorithm 1, and it is $O(K_S \times (\Pi_i K_i)^{L+1})$ where $K_S = |\Sigma| \times |S|\,!$, $K_i = |\Sigma'_i|^{\,|Q'_i|}$, and $L$ is the maximal length of a path in the synchronization graph $\mathcal{S}$. A possible way to limit the complexity is to impose the following restriction on modular automata.

**Definition 22** *A modular automaton $MA = ((A'_i)_{i=1}^{n}, \mathcal{S})$ is* quasi-reversible *if, for any synchronized transition $\delta(s, \langle \phi, \sigma \rangle) = s'$ in $\mathcal{S}$ and any $i \in dom(\sigma)$, either $s$ and $s'$ occur on some cycle of $\mathcal{S}$ or $s(i)$ and $\phi(i)$ occur jointly on some cycle of $A'_i$, or both.*

For quasi-reversible automata, all sets $PIP(s_{j-1}(i), \phi_j(i))$ are singletons, and the number of linear systems to be solved decreases to $O(K_S \times (\Pi_i K_i))$. This is still a large number in comparison with the number of equations in each system, which is $O(K_S + (\sum_i K_i))$, but time complexity is in some sense less dramatic than space complexity.

# 7 Conclusion

This paper has presented a first attempt to synthesize P/T-nets from symbolic transition systems up to language equivalence. The symbolic transition systems which have been considered are the modular automata, halfway between indexed families of automata and their products. In order to be able to compute the regions of a modular automaton within reasonable space, we restricted ourselves to distributed P/T-nets and regions. The distribution constraints on P/T-nets are inherited from modular automata. In order to be able to decide on the synthesis problem within reasonable time, we suggested to impose on modular automata the constraint of quasi-reversibility, meaning that whenever one can jump from $\langle q_1, \ldots, q_n \rangle$ to $\langle q'_1, \ldots, q'_n \rangle$ under the effect of a synchronized transition, one can come back to $\langle q_1, \ldots, q_n \rangle$ by some sequence of local or synchronized transitions. Stronger forms of reversibility are frequently considered in the field of Discrete Event Systems. We hope that the algorithms defined in this paper will allow extending the application of distributed P/T-net synthesis to automata significantly larger than the non-modular automata dealt with at present with Synet (at most $10^4$ states). On a less practical side, a possible extension of this work is to cope with modular automata with more than two levels.

# References


[1] Badouel, E., Bernardinello, L., Darondeau, Ph.: Polynomial Algorithms for the Synthesis of Bounded Nets. Proc. CAAP, LNCS **915** (1995) 647-679.

[2] Badouel, E., Caillaud, B., Darondeau, Ph.: Distributing finite automata through Petri net synthesis. *Formal Aspects of Computing* **13** (2002) 447-470.

[3] Bernardinello, L., De Michelis, G., Petruni, K., Vigna, S.: On the Synchronic Sructure of Transition Systems. In J. Desel, editor, *Structures in Concurrency Theory*, Workshops on Computing, Springer Verlag (1996) 11-31.

[4] Bernardinello, L.: Synthesis of Net Systems. Proc. ATPN, LNCS **691** (1993) 11-31.

[5] Cortadella, J., Kishinevsky, M., Lavagno, L., Yakovlev, A.: Deriving Petri Nets from Finite Transition Systems. *IEEE Trans. on Computers* **47**,8 (1998) 859-882.

[6] Christensen, S., Petrucci, L.: Modular State Space Analysis of Coloured Petri nets. Proc. ATPN, LNCS **935** (1995) 201-217.

[7] Darondeau, P.: Region Based Synthesis of P/T-nets and its Potential Applications. Proc. ATPN, LNCS **1825** (2000) 16-23.

[8] Desel, J., Reisig, W.: The Synthesis Problem of Petri Nets. *Acta Informatica* **33** (1996) 297-315.

[9] Duboc, C.: Mixed Product and Asynchronous Automata. *Theoretical Computer Science* **48**, 3 (1986) 183-199.

[10] Ehrenfeucht, A., Rozenberg, G.: Partial (Set) 2-Structures; *Part I:* Basic Notions and the Representation Problem. *Acta Informatica* **27** (1990) 315-342.

[11] Ehrenfeucht, A., Rozenberg, G.: Partial (Set) 2-Structures; *Part II:* State Spaces of Concurrent Systems. *Acta Informatica* **27** (1990) 343-368.

[12] Ghaffari, A.: Les réseaux de Petri pour la synthèse de contrôle des systèmes à événements discrets. PhD thesis, Univ. of Metz (2002).

[13] Ghaffari, A., Rezg, N., Xie, X.: Design of Live and Maximally Permissive Petri Net Controller Using Theory of Regions. *IEEE Trans. on Robotics and Automation* **19**, 1 (2003) 137-142.

[14] Mukund, M.: Petri Nets and Step Transition Systems. *International Journal of Foundations of Computer Science* **3**,4 (1992) 443-478.

[15] Petrucci, L.: Modélisation, vérification et applications. Mémoire d'habilitation à diriger des recherches, Université d'Evry (2002).

[16] http://www.irisa.fr/s4/tools/synet/

# 8 Appendix

**Lemma 1.** *$\forall s \in S\ \exists w \in \Sigma^*\ \delta(\langle s_0, s_0\rangle, w) = \langle s, s\rangle$ in $Exp(MA)$.*

*Proof.* Straightforward from definitions 7 and 8. ∎

**Lemma 2.** *If $q \xrightarrow{w} q' \in Exp(MA)$ and $w \equiv w'$, then $q \xrightarrow{w'} q' \in Exp(MA)$.*

*Proof.* Let $\sigma' \in \Sigma'_i$ and $\sigma \notin \Sigma_i$. We show that $\delta(\langle\vec{q}, s\rangle, \sigma\sigma') = \delta(\langle\vec{q}, s\rangle, \sigma'\sigma)$ for any state $\langle\vec{q}, s\rangle$ of $Exp(MA)$. When $\sigma \in \Sigma'_j$ for some $j$ (hence $j \neq i$) the proof is immediate using twice condition (i) in Def. 8 for $\sigma$ and $\sigma'$ or conversely. We assume therefore that $\sigma \in \Sigma \setminus \Sigma'$, and we let $\vec{q} = \langle q_1, \dots, q_n\rangle$. If $\delta(\langle\vec{q}, s\rangle, \sigma'\sigma)$ is defined, then by (i) in Def. 8, $\delta(\langle\vec{q}, s\rangle, \sigma') = (\langle q_1, \dots, \delta'_i(q_i, \sigma'), \dots, q_n\rangle, s)$, and by (ii) in Def. 8, $s \xrightarrow{\langle\phi,\sigma\rangle} s'$ in $\mathcal{S}$ for some $\phi$ such that $q_j = \phi(j)$ for all $j \in dom(\sigma)$. Let $\delta(\langle\vec{q}, s\rangle, \sigma'\sigma) = \langle\vec{q'}, s'\rangle$. As $i \notin dom(\sigma)$, $\vec{q'} = \langle q'_1, \dots, q'_n\rangle$ is the state vector defined with $q'_i = \delta'_i(q_i, \sigma')$, $q'_j = s'(j)$ for $j \in dom(\sigma)$, and $q'_j = q_j$ otherwise. As $i \notin dom(\sigma)$, by (ii) in Def. 8, $\delta(\langle\vec{q}, s\rangle, \sigma)$ is defined and $\delta(\langle\vec{q}, s\rangle, \sigma) = \langle\vec{q''}, s'\rangle$ with $q''_j = s'(j)$ for $j \in dom(\sigma)$, and $q''_j = q_j$ otherwise. In particular $q''_i = q_i$, hence $\delta'_i(q''_i, \sigma)$ is defined. By (i) in Def. 8, $\delta(\langle\vec{q}, s\rangle, \sigma'\sigma)$ is defined and it is equal to $\langle\vec{q'}, s'\rangle$. One can prove in a similar way that whenever $\delta(\langle\vec{q}, s\rangle, \sigma\sigma')$ is defined, $\delta(\langle\vec{q}, s\rangle, \sigma'\sigma)$ is defined and they are equal. ∎

**Lemma 3.** *$\forall s, s' \in S\ \forall w \in L(s, s')\ \delta(\langle s, s\rangle, w) = \langle s', s'\rangle$ in $Exp(MA)$.*

*Proof.* Straightforward from definitions 7 and 8. ∎

**Proposition 4.** *$R(MA)$ is a minimal set of representatives for $L(MA)$ w.r.t. the congruence $\equiv$ and it is a regular language.*

*Proof.* All modules $A'_i$ of $MA$ have finite sets of states $Q'_i$, hence the languages $L(s, s')$ and $L(s)$ are regular for all $s$ and $s'$. As the synchronization graph has a finite set of vertices $S$, it follows from the definition of $L^+(s, s')$ that $R(MA)$ is regular. The inclusion $R(MA) \subseteq L(MA)$ is obvious from the definition of the expansion of a modular automaton (Def. 8). It should be clear from the definition of $\equiv$ that two congruent words of $L(s, s')$ or $L(s)$ must be equal, and a similar property holds for $R(MA)$ (two synchronized actions in $\Sigma \setminus \Sigma'$ never commute). It remains to show that every word $w \in L(MA)$ is represented by some congruent word $w' \in R(MA)$. Take any word $w$ in $L(MA)$. Since $\varsigma_i\varsigma_j \equiv \varsigma_j\varsigma_i$ for all $\varsigma_i \in \Sigma'_i$ and $\varsigma_j \in \Sigma'_j$ such that $i \neq j$, $w \equiv w''$ for some $w''$ in the intersection of $L(MA)$ and the set $((\Sigma_1'^* \dots \Sigma_n'^*)(\Sigma \setminus \Sigma'))^*(\Sigma_1'^* \dots \Sigma_n'^*)$. As $\varsigma_i\sigma \equiv \sigma\varsigma_i$ whenever $\sigma \in \Sigma \setminus \Sigma'$ and $\varsigma_i \in \Sigma'_i$ with $i \notin dom(\sigma)$, $w'' \equiv w'$ for some $w'$ in $R(MA)$, obtained pushing repeatedly to the right all local actions $\varsigma_i \in \Sigma'_i$ that commute with the next synchronized action $\sigma \in \Sigma \setminus \Sigma'$ on their right (hence such that $i \notin dom(\sigma)$). ∎

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifi que,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-0803